

METHOD FOR TRANSMITTING MEDIA FILES OVER COMMUNICATION NETWORK

Publication number: RU2189119 (C2)

Publication date: 2002-09-10

Inventor(s): EHVERETT STJUART NIL [GB]; BARRS DZHEJMS [GB];
BUD DZHEJMS AVGUSTIN [GB]; MONTGOMERI JAN
DEHVID [GB]

Applicant(s): INSIGMA TEKHOLODZHIZ LTD [GB]

Classification:

- international: **G06F9/445; G06F12/00; G06F13/00; G10H1/00;
G10K15/02; G10L19/00; H04L29/06; H04N7/173;
H04N7/24; H04W80/00; G06F9/445; G06F12/00;
G06F13/00; G10H1/00; G10K15/02; G10L19/00;
H04L29/06; H04N7/173; H04N7/24; H04W80/00; (IPC1-
7): H04N7/173; H04Q7/00**

- European: H04N7/173B; H04N7/24T6

Application number: RU19990116367 19980126

Priority number(s): NZ19970314150 19970129

Also published as:

WO9833320 (A1)

ZA9800690 (A)

US2002120675 (A1)

JP2001509280 (T)

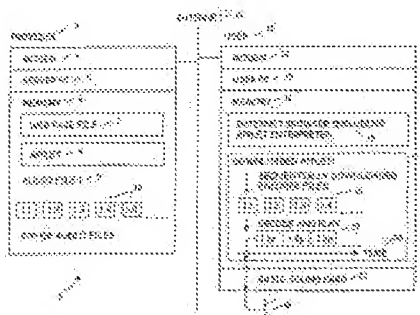
EP0962097 (A1)

more >>

Abstract of RU 2189119 (C2)

transmission and reproduction of media files over communication network in predetermined sequence.

SUBSTANCE: method for transmitting media files over communication network, primarily through the Internet, involves separation of files in provider computer into sequence of coded files which are handled in provider computer and conveyed over communication network in definite sequence to receiving computer. User program being loaded is also stored in provider computer and transmitted together with all types of files constituting message content to receiving computer over communication network. Files obtained may be then reproduced by receiving computer in precise sequence. EFFECT: enlarged functional capabilities. 9 cl, 9 dwg



.....
Data supplied from the **esp@cenet** database — Worldwide



(19) **RU** ⁽¹¹⁾ **2 189 119** ⁽¹³⁾ **C2**
(51) МПК⁷ **H 04 N 7/173, H 04 Q 7/00**

РОССИЙСКОЕ АГЕНТСТВО
ПО ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

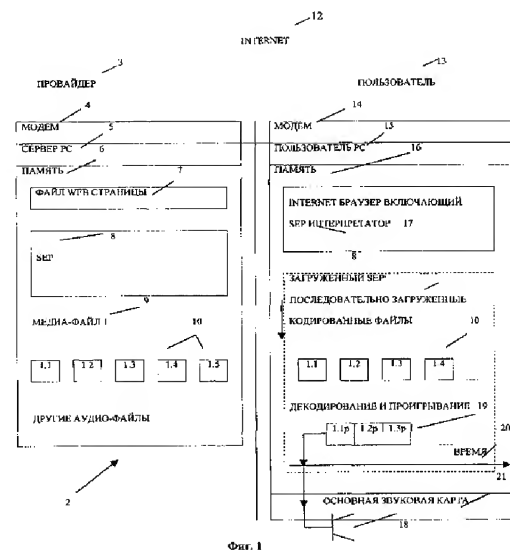
(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ РОССИЙСКОЙ ФЕДЕРАЦИИ

(21), (22) Заявка: 99116367/09, 26.01.1998
(24) Дата начала действия патента: 26.01.1998
(30) Приоритет: 29.01.1997 NZ 314150
(43) Дата публикации заявки: 20.10.2001
(46) Дата публикации: 10.09.2002
(56) Ссылки: EP 0702493 A1, 20.03.1996. RU 2050695 C1, 20.12.1995. RU 2051472 C1, 27.12.1995. WO 9504431 A2, 09.02.1995. US 5440336 A, 08.08.1995.
(85) Дата перевода заявки РСТ на национальную фазу: 30.08.1999
(86) Заявка РСТ: NZ 98/00005 (26.01.1998)
(87) Публикация РСТ: WO 98/33320 (30.07.1998)
(98) Адрес для переписки: 113054, Москва, Б.Строchenовский пер. 22/25, Фирма "Бейкер и Макензи" для Е.А.Ариевича, пат.пов. В.П.Зылю, рег № 93

(71) Заявитель:
ИНСИГМА ТЕХНОЛОДЖИЗ ЛИМИТЕД (GB)
(72) Изобретатель: ЭВЕРЕТТ Стюарт Нил (GB),
БАРРС Джеймс (GB), БУД Джеймс Августин (GB), МОНТГОМЕРИ Ян Дэвид (GB)
(73) Патентообладатель:
ИНСИГМА ТЕХНОЛОДЖИЗ ЛИМИТЕД (GB)
(74) Патентный поверенный:
Зыль Валерий Петрович

(54) СПОСОБ ПЕРЕДАЧИ МЕДИА-ФАЙЛОВ ПО СЕТИ СВЯЗИ

(57) Изобретение относится к способу передачи и воспроизведения медиа-файлов по сети связи в заранее определенной последовательности. Техническим результатом является расширение функциональных возможностей. В способе передачи медиа-файлов по сети связи, в типичном случае, через Интернет, файлы разделяются в компьютере провайдера на последовательность кодированных файлов, которые ведутся в компьютере провайдера и передаются по сети связи в определенной последовательности к принимающему компьютеру. Загружаемая программа пользователя также хранится в компьютере провайдера и вместе со всеми типами файлов, составляющих контент сообщения, также передается по сети связи к принимающему компьютеру. Полученные файлы могут быть затем воспроизведены принимающим компьютером в точной последовательности. 2 с. и 7 з.п. ф-лы, 9 ил.





(12) ABSTRACT OF INVENTION

(71) Applicant:
INSIGMA TEKhNOLODZhIZ LIMITED (GB)

(72) Inventor: EhVERETT Stjuart Nil (GB),
BARRS Dzhejms (GB), BUD Dzhejms Avgustin
(GB), MONTGOMERI Jan Dehvid (GB)

(73) Proprietor:
INSIGMA TEKhNOLODZhIZ LIMITED (GB)

(74) Representative:
Zvl' Valerii Petrovich

(57) Abstract:

12

ИНТЕРНЕТ

13

ПОЛЬЗОВАТЕЛЬ

14

МОДЕМ

15

ПОЛЬЗОВАТЕЛЬСКИЙ РС

16

ПАМЯТЬ

17

ИНТЕРНЕТ БРАУЗЕР ВКЛЮЧАЮЩИЙ
СЕР ИНТЕРПРЕТАТОР

18

ЗАГРУЖЕННЫЙ WEB

19

ПОСЛЕДОВАТЕЛЬНО ЗАГРУЖЕННЫЕ
КОДИРОВАННЫЕ ФАЙЛЫ

20

1.1 1.2 1.3 1.4

21

ДЕКОДИРОВАНИЕ И ПРОИГРЫВАНИЕ

22

ВРЕМЯ

23

ОСНОВНАЯ ЗВУКОВАЯ КАРТА

24

1.1п 1.2п 1.3п

25

МОДЕМ

26

СЕРВЕР РС

27

ПАМЯТЬ

28

ФАЙЛ ВПЕ СТРАНИЦЫ

29

WEB

30

МЕТДА-ФАЙЛ

31

1.1 1.2 1.3 1.4 1.5

32

ДРУГИЕ АУДИО-ФАЙЛЫ

33

2

Фиг. 1

Настоящее изобретение относится к способу передачи и воспроизведения медиа и других типов файлов по сети связи в заранее определенной последовательности/порядке и управляемого воспроизведения файлов упомянутых типов в соответствии с упорядочивающей временной шкалой, при котором воспроизведение интеллектуально управляется в соответствии с колебаниями пропускной способности сети и тем влиянием, которое они оказывают на доступность последующих файлов. В частности, но не только настоящее изобретение относится к компьютеризированному способу передачи и воспроизведения медиа или других типов файлов по Internet, где эти типы файлов являются составными частями более крупных композиций, и воспроизведение упомянутых файлов происходит в процессе реконструкции "на лету" более крупных композиций программой Internet-браузера.

В отношении медиа-файлов существует несколько способов их передачи с помощью компьютеров по сетям связи, таким как Internet, но все они имеют определенные недостатки. Некоторые известные методы загрузки аудиофайлов основаны на отправке аудиофайла в виде пакета данных или оцифрованного звука, и приемник должен ожидать пока медиа-файл не загрузится полностью, прежде чем начнется полное воспроизведение файла. В частности, применительно к Internet, продолжительность времени, требующегося для загрузки медиа (MIDI-файла [файл в стандарте MIDI (Musical Instrument Digital Interface цифровой интерфейс музыкальных инструментов)]) может быть настолько продолжительным, что пользователь может принять решение отключиться до того, как медиа-файл будет загружен и сможет быть проигран.

При использовании способов потоковой передачи медиа-файлов в режиме реального времени, таких как, например, RealAudio или Shockwave, происходит перенос сжатых аудиофайлов, которые декодируются и проигрываются по мере получения их приемником с использованием метода буферизации. Однако эти способы передачи сжатых аудиофайлов обычно требуют стабильной полосы пропускания и соответствующего программного обеспечения на стороне приемника. Другим недостатком систем RealAudio является то, что для адекватного воспроизведения звука они требуют модема, имеющего скорость как минимум 28,8 Кбит/с. Такие системы обычно и лучше всего использовать для вещания радиостанций, трансляции концертов или для видеоконференций, но они не годятся для быстрого встраивания в узлы глобальной гипертекстовой сети (World Wide Web) Internet, поскольку при доступных в настоящее время значениях полосы пропускания, Internet в целом не может поддерживать много одновременных потоков непрерывных аудио-/видеоданных, исходящих и большого числа Web-узлов. Решения, основанные на потоковой передаче медиа в режиме реального времени, предоставляют определенные возможности разработчику Internet, но у них отсутствует способность возобновления проигрывания, взаимодействия с пользователем, отслеживания статистики и использования

доступной полосы пропускания.

Flash фирмы Macromedia является новым и распространенным способом введения развлекательных мультимедиа в Web-узлы и использует понятие временной шкалы для проигрывания комбинации файлов изображения и звука. Однако Flash в очень большой мере основывается на использовании векторной графики и очень малых аудиофайлов. Поскольку Flash композиции в значительной мере должны быть предустановлены в браузер перед запуском, они должны иметь минимальный размер и неспособны интеллектуально продолжать сборку в браузере. Web-узлы Flash производят довольно сходное впечатление в силу природы векторных изображений и, как таковые, пригодны только для некоторых видов кампаний в Internet и не вполне подходят для встраивания в существующие Web-узлы.

SMIL (Synchronized Multimedia Integration Language) язык синхронизированной интеграции мультимедиа является новым открытым стандартом Internet и позволяет вызывать объекты в программу браузера в заданных точках. Это является одной из простейших форм упорядочения и требует достаточно новых версий браузера для поддержки такой технологии. SMIL не может обеспечить какой-либо гибкости в отношении предварительной выборки, многоуровневости и, что более важно, интеллектуального воспроизведения композиций в соответствии с колебаниями пропускной способности сети. SMIL был создан с расчетом на концепцию широкополосной технологии, в то время как скорость передачи сообщений SMIL в современных узкополосных сетях тормозится вследствие большого времени, которое требуется для извлечения требуемого файла в заданной точке.

Методы MIDI также используются для генерации звука. По своей природе MIDI-файлы имеют меньший размер, чем файлы, которые пытаются сохранять подлинную волновую картину звука в цифровом формате. Это означает, что они могут быть легко переданы по сети, быстрее, чем другие типы аудиофайлов. Однако MIDI-файлы не воспроизводят заранее записанные звуки. Вместо этого компьютер, в соответствии со стандартом, известным как GM MIDI, исполняет набор инструкций с помощью звуковой карты и активирует ноты того или другого инструмента, приблизительные звуковые характеристики которого были сохранены в этой звуковой карте. Качество звуковой карты или устройства, связанного со звуковой картой и способного воспринимать набор инструкций GM MIDI, сильно варьирует и в значительной мере определяется его ценой. Следовательно, для получения реалистичного звучания необходимы дорогостоящие аппаратные блоки. Качество воспроизведения аудиофайлов обычно низкое, что обусловлено природой FM синтеза в массовых звуковых картах низкого класса. Даже с высококлассными звуковыми картами качество воспроизведения аудиофайлов ограничивается предустановленными параметрами GM MIDI, и такие установки дают возможность воспроизводить только основные звуки

инструментов, что пригодно только для ограниченного круга приложений компьютерных игр и тому подобное. MIDI-файлы основываются на упорядочивающей временной шкале, но не требуют какого-либо интеллектуального управления для адаптации к изменениям пропускной способности сети, поскольку такая последовательность вызывает те звуки, которые уже встроены в звуковую карту компьютера, а не медиа-файлы, которые требуют загрузки.

Ни один из вышеупомянутых методов не пытается решить задачу передачи мультимедиа или данных по сети связи, где медиа-данные разделены на сегменты для передачи их по сети и затем реконструируются в составной отрывок контента [цифрового представления медиа-данных], который строится по кускам во время посещения Web-узла, с использованием имеющейся полосы пропускания для предварительной загрузки сегментов готовых для повторного проигрывания в соответствии с управляемой по времени последовательностью. Эта последовательность определяет построение композиции внутри браузера, где избранные сегменты могут быть вновь использованы или для повторения в соответствии со стилем отрывка, или повторены циклически, или воспроизведены позже при повторном проигрывании, которое является непрерывным или управляемым по времени. Механизм повторного проигрывания содержит специальные резервные события, которые позволяют повторно использовать избранные файлы в том случае, если последующий файл недоступен вследствие перегрузки сети, и дополнительные резервные события, что позволяет изменять свойства избранных файлов в зависимости от состояний загрузки и использования резервного события "повторное использование файла". Такой способ упорядочивания позволяет вводить и использовать "время" в таком виде сообщения как Web-узел с тем, чтобы производить запуск контента через определенный промежуток времени и в ситуации, когда файлы уже предварительно загружены.

Отсюда очевидно, что имеется необходимость в системе, которая была бы способна передавать и проигрывать или воспроизводить медиа-файлы через сеть связи, такую как Internet, которая была бы совместима с обеспечением Web-страниц и которая обеспечивала бы интеллектуальную синхронизацию и планирование во времени файлов для того, чтобы обеспечить построение контента из составляющих его файлов, по мере того, как они предварительно загружаются в последовательном порядке, и при которой время доступа, определяемое по времени ожидания начала проигрывания медиа-файла на компьютерном терминале пользователя, сокращается за счет того, что этой системе, для того чтобы начать проигрывание, требуется присутствие только одного файла или некоторой выборки файлов.

Также очевидно, что имеется необходимость в системе, которая была бы способна передавать и воспроизводить через сеть связи, такую как Internet, все типы

файлов, и которая оптимальным образом использовала бы имеющуюся полосу пропускания, позволяя упорядочивать и управлять загрузкой всех упомянутых файлов с помощью небольшой исполняемой программы и устанавливать инструкции для упорядочивания, которые определяют порядок загрузки всего контента сообщения и порядок его воспроизведения, при котором сообщение представляет собой голос за кадром, музыкальную композицию или другую фонограмму, аудиовизуальный отрывок, содержащий неподвижные изображения или анимационные изображения, синхронизованные со звуком или другим образом, а также взаимодействие с пользователем или другим образом, отрывок потоковых медиа или других мультимедийных компонентов, таких как Flash фильмов и других композиций, таких как компоненты, требующиеся для Web-узла. Таким контентом могут быть файлы HTML [hypertext markup language язык разметки гипертекста] или файлы подобного типа, текстовые файлы, файлы изображений, мультимедийные файлы и аудиофайлы.

Также очевидно, что имеется необходимость в системе, которая была бы способна передавать мультимедиа или данные по сети связи, в которой медиа-данные разделены на сегменты для передачи по сети и затем реконструируются в составной отрывок контента, такой как музыкальная композиция или аудиовизуальный отрывок, которые строятся по кускам и во время визита на Web-узел с использованием любой имеющийся полосы пропускания для предварительной загрузки сегментов, готовых для повторного проигрывания в соответствии с управляемой по времени последовательностью. Эта последовательность определяет построение композиции внутри браузера, где избранные сегменты могут быть использованы повторно, или для повторения в соответствии со стилем отрывка, или повторены циклически, или воспроизведены позже при повторном проигрывании, которое является непрерывным или управляемым по времени. Механизм повторного проигрывания содержит специальные резервные события, которые позволяют повторно использовать избранные файлы в том случае, если последующий файл недоступен вследствие перегрузки сети, и дополнительные резервные события, которые позволяют изменять свойства избранных файлов в зависимости от состояний загрузки и использования резервного события "повторное использование файла". Такой способ упорядочивания позволяет вводить и использовать "время" в таком виде сообщения, как Web-узел, с тем, чтобы производить запуск контента через определенный промежуток времени и в ситуации, когда файлы уже предварительно загружены.

Целью настоящего изобретения является создание интеллектуального механизма доставки, который обеспечивает такой способ передачи и проигрывания медиа-файлов через сеть связи, который позволяет устранить или по крайней мере свести к минимуму вышеупомянутые недостатки.

Целью настоящего изобретения является также создание интеллектуального

инструмента доставки, который обеспечивает такой способ передачи всех типов файлов через сеть связи, и воспроизведения упомянутых файлов таким образом, чтобы обеспечить практически осуществимое использование мультимедиа в сетях, где пропускная способность сильно варьирует и имеется потребность в возможности возобновления использования, а также интеллектуального повторного использования и управления файлами и соответствующей регулировки их свойств с тем, чтобы построить и доставить полный отрывок контекста или целое сообщение с помощью синхронизованной доставки, который устраняет или по крайней мере сводит к минимуму недостатки существующих способов передачи данных.

В общих чертах, изобретение содержит способ передачи медиа-файла или другого типа файла по сети связи и воспроизведения или проигрывания упомянутых файлов, который включает:

(a) разделение медиа-контента на последовательность сегментированных файлов,

(b) поддержание упомянутых сегментированных файлов в средствах компьютера провайдера и

(c) передача упомянутых сегментированных файлов к средствам принимающего компьютера в определенной последовательности,

в котором, после того как каждый сегмент был получен упомянутым принимающим компьютером, проигрывание или воспроизведение сегмента будет начинаться до или во время загрузки следующего последовательного сегмента, и, кроме того, в котором последующие полученные сегменты будут проигрываться в соответствии с информацией об очередности, в котором информация об очередности определяет, проигрывается ли какой-либо сегмент, какой именно и каким образом, в том случае, когда следующий сегмент последовательности не доступен для проигрывания или воспроизведения.

Предпочтительно стадия (b), кроме того, включает поддержание загружаемой пользователем программы в средствах компьютера провайдера и стадия (c), кроме того, включает в себя передачу этой программы к устройствам принимающего компьютера.

Предпочтительно сегмент кодируется и декодируется загружаемой пользователем программы перед проигрыванием или воспроизведением в средствах принимающего компьютера с помощью.

Предпочтительно эта программа является Java-апплетом [приложением] или Active X файлом.

Предпочтительно проигрывание медиа-файла может начинаться до завершения процесса получения следующего сегмента последовательности.

Предпочтительно информация об очередности параметризована и передается в виде текстового документа или другого подобного типа документа.

Предпочтительно информация об очередности встроена в загружаемую пользователем программу.

Предпочтительно медиа-файл

представляет собой любую совокупность данных, таких как аудиофайл, видеофайл, файл изображения, файл анимационного изображения, HTML файл, VRML/3d World [Virtual Reality Modeling Language язык моделирования виртуальной реальности трехмерного мира] файл, текстовый файл. Flash фильм, другие исполняемые файлы, инструкции для других устройств, потоки данных или фильтр, который модифицирует другие медиа.

В другом варианте изобретение предполагает также наличие системы принимающего компьютера, включающей средства для воспроизведения или проигрывания медиа-файла, переданного по сети связи от средств компьютера провайдера, имеющего средства для разделения и поддержания медиа-файл в виде последовательности сегментов и передачи этих сегментов в определенной последовательности по сети связи к средствам принимающего компьютера, в котором после того как каждый упомянутый сегмент был получен упомянутым принимающим компьютером, он декодируется, и проигрывание или воспроизведение упомянутого сегмента может начинаться до или во время загрузки следующего сегмента последовательности таким образом, что декодированные файлы могут быть воспроизведены или проиграны способом, идентичным тому, которым была построена композиция или последовательность до того как произошло какое-либо резервное событие в ходе воспроизведения.

Предпочтительно изобретение предполагает способ передачи медиа-файла по сети связи в соответствии с одним или более из следующих примеров.

С помощью прилагаемых чертежей описан один из предпочтительных способов передачи и воспроизведения медиа- или файла другого типа по сети связи, в котором:

фиг.1 - блок-схема предпочтительного способа работы данной системы;

фиг.2 - диаграмма работы основной системы;

фиг.3 - образец псевдокода для апплета предпочтительного варианта;

фиг. 4 - диаграмма, которая схематически иллюстрирует одну из стадий в повторном проигрывании сегмента медиа-файла;

фиг. 5 - диаграмма, которая схематически иллюстрирует еще одну стадию в повторном проигрывании сегмента медиа-файла;

фиг. 6 - диаграмма, которая схематически иллюстрирует еще одну стадию в повторном проигрывании сегмента медиа-файла;

фиг. 7 - диаграмма, которая схематически иллюстрирует другую стадию в повторном проигрывании сегмента медиа-файла;

фиг. 8 - диаграмма, которая схематически иллюстрирует другую стадию в повторном проигрывании сегмента медиа-файла;

фиг. 9 - диаграмма, которая схематически иллюстрирует другую стадию в повторном проигрывании сегмента медиа-файла.

фиг.4 - 9 в комбинации иллюстрируют интеллектуальную природу метода.

Система 2, как показано на общей блок-схеме фиг.1, содержит провайдер 3, включающий модем 4 провайдера, сервер 5 и память 6 провайдера, которая содержит

Web-страницу 7, небольшую исполняемую программу [SEP] 8 и один или более медиа-файлов 9, разделенных на сегменты 10, длина и размер которых определяются разработчиком или диктуются природой исходных медиа, в которых, например, музыка обычно может быть разделена на логичные фразы.

Эти сегменты 10 создаются как индивидуальные файлы, которые имеют смысл при проигрывании их индивидуально или в сочетании с другим файлом, как часть целой пьесы или многоуровневой композиции. Так же как и короткая музыкальная фраза, произносимая речь может быть разделена на отдельные слова, предложения или фразы, а анимации могут быть разделены на фоновые изображения и изображения переднего плана.

Сегмент 10 может быть использован повторно в любое время в ходе последовательности. Примером этого могут служить музыкальные такты барабанов, которые отбивают ритм, или фоновое изображение в анимации. С целью экономии ресурсов, эти многократно используемые сегменты 10 размещаются в компьютере пользователя. Применительно к Internet этими ресурсами могут быть полоса пропускания и время загрузки, где SEP оптимально использует полосу пропускания в любое время до тех пор, пока не будут переданы все сегментированные файлы.

Размещение этих многократно используемых сегментов 10 в компьютере пользователя происходит также с целью регулировки скорости синхронизации для управления воспроизведением сегментированных файлов должным образом, но, по умолчанию, эта синхронизация может быть использована для управления объявлениями и другими повторяющимися во времени событиями и/или контентом, что вводит истинное понятие "временной развертки просмотра" для таких объектов, как Internet Web-узел или другие сообщения.

Провайдер 3 может быть соединен с множеством пользователей 13 сетью связи 12, такой как Internet. Система 13 пользователя включает модем 14 пользователя или другое устройство подключения к сети, компьютер 15 пользователя и память 16 пользователя, содержащую Internet браузер 17. Браузер 17 содержит интерпретатор, который интерпретирует и исполняет SEP 8 и его инструкции относительно очередности.

Как показано на фиг.1 и 2, компьютер провайдера поддерживает в памяти 6 провайдера, файл 7 Web-страницы, SEP 8 и один или более медиа-файлов 9, или другие файлы в виде кодированных файлов или других типов файлов 10, которыми могут быть аудио, видео, графические, HTML и другие известные типы файлов, которые составляют контент сообщения по сети связи. SEP 8 получает от сервера 5 кодированные файлы или другие типы файлов, которые представляют собой медиа-файл 9, полный контент, музыкальный фрагмент или другую композицию, в виде некоторого числа частей, каждая из которых закодирована или сжата в кодированный файл.

Система 13 пользователя может загрузить Web-страницу 7, SEP 8 и набор инструкций провайдера 3, определяющих очередность,

используя Internet 12, а также модемы пользователя и провайдера или другие интерфейсы устройств подключения к сети 14 и 4 соответственно. Web-страница 7 обеспечивает пользователю 13 возможность загрузки и воспроизведения одного или более медиа-файлов 9. При выборе файла 9 SEP 8, которая теперь находится в памяти пользователя (обведена штриховой линией), исполняется программой браузера 17 пользователя. SEP 8 предпочтительно написана на Java или ActiveX для исходной цели как приложение Internet, хотя может быть использован любой язык, интерпретируемый браузером 17, если только он поддерживает медиа-файлы. В другом варианте пользователь 13 может использовать резидентную исполняемую прикладную программу для загрузки и воспроизведения кодированных файлов 10.

В другом варианте исходная Web-страница 7 и SEP 8 провайдера 3 могут быть загружены и, как только будет загружен SEP 8, исполняемый программой браузера 17, он может управлять дальнейшей загрузкой всех типов файлов, медиа или других, которые образуют весь контент сообщения (Web-страницы и их контент).

SEP 8 начинает загружать первый из последовательных кодированных файлов или других типов файлов 10 [1.1] и ожидает пока он полностью загрузится в память 16 компьютера пользователя. Затем SEP 8 декодирует или разворачивает кодированный файл 10 [1.1] в декодированный файл 19 [1.1 p] и начинает проигрывание или воспроизведение файла, если это предписано набором инструкций очередности. Затем, как можно раньше, обычно вместе с проигрыванием загруженных файлов SBP 8 начинает загружать следующий из последовательности кодированных файлов или файлов другого типа 10 [1.2] [1.3] и по завершении загрузки каждого кодированного файла или другого типа файла 10 SEP 8 декодирует его в декодированный файл 19 и может начинать проигрывание или воспроизведение его в той точке, которая предписывается информацией об очередности. На практике декодированные файлы 19 [1.1p], [1.2p]... будут добавляться в очередь, которая делает возможным проигрывание каждого декодированного файла 19 в порядке "первый вошел первый вышел (FIFO)", так что один файл, например, [1.1p] переходит в другой [1.2p], если это предписывается информацией об очередности. Таким образом, будет создаваться впечатление, что медиа-файл или другой тип файла 9 проигрывается непрерывно без пауз между декодированными файлами. Следовательно, эта система требует от пользователя 13 наличия только основной звуковой карты 21 (с целью воспроизведения звука) и модема 14 или другого устройства подключения к сети для проигрывания высококачественных медиа (аудио) файла 9 или других типов файлов.

В другом варианте файлы могут быть расположены для проигрывания или воспроизведения в соответствии с последовательностью, в которой используется спланированный во времени перерыв. В таком случае следующий файл 19 из очереди будет загружен при первой же

возможности и останется в памяти 16 компьютера пользователя до того времени, которое указано в информации об очередности. Информация об очередности может быть параметризована в текстовом документе или встроена в исполняемую программу.

Распределение по времени загрузки кодированных файлов или других типов файлов 10 и проигрывание декодированных файлов 19 видно на линии 20 фиг.1, где первый декодированный файл [1.1p] не начинает проигрываться до тех пор, пока соответствующий ему кодированный файл 10 [1.1] не будет полностью загружен и кодирован. В информации об очередности определены точки циклического возврата, которые указывают, что соответствующие фразы должны повторяться в том случае, если дальнейшее прохождение последовательности невозможно (обычно, хотя не обязательно, потому что требуемый медиа-файл или другой тип файла 10 еще не полностью загружен 10 или декодирован 19).

Наличие и доступность точек циклического возврата создает впечатление непрерывного вывода с целью достижения непрерывного потока. Кроме точек циклического возврата, используются также другие резервные события и инструкции для большей управляемости воспроизведением файлов и, в частности, для поддержания непрерывного воспроизведения. Команды "проигрывать один раз" применяются к файлам, которые не должны повторяться в случае, если событие, запускающее цикл, произойдет в области, где находится упомянутый файл. "Проигрывать один раз" используется для таких файлов как речевые файлы, где повторение речи в цикле будет выглядеть странно. Команда "требуется для" применяется для того, чтобы сгруппировать вместе более чем один файл для сохранения точной фразировки.

"Требуется для" может также применяться для инициализации точки циклического возврата в положении, которое более подходит для данной композиции, чем точка запуска по умолчанию, определенная положением "играть" в отсутствующем файле.

Пользователь 13 должен только дожидаться загрузки и декодирования первого кодированного файла или другого типа файла 10 [1.1] и ему не нужно ждать загрузки всего медиа, всего контента, музыкального отрывка или другой композиции, или других типов файлов прежде, чем медиа-файл 9 или другой тип файла начнет проигрываться или воспроизводиться. Разделение всего медиа-файла избавляет от необходимости загрузки всего медиа-файла до того, как он начнет проигрываться.

SEP 8, предпочтительно написанный на Java, содержит два элемента или "цепочки", которые исполняются одновременно во время жизни Java SEP (как показано на фиг.3). Первый элемент представляет собой ядро или загрузчик, который запускает второй элемент и загружает кодированные файлы 10 от провайдера 3. Второй элемент представляет собой проигрыватель/программатор, который находится в цикле и непрерывно отслеживает состояние кодированных и декодированных файлов 10 и 19. В течение цикла, если проигрыватель/программатор обнаруживает,

что декодированный файл доступен для проигрывания (т.е. и кодированный файл был загружен и декодирован), он проигрывает этот файл (в начале цикла, чтобы сохранить синхронизацию), коль скоро этот файл подчиняется определенному для него набору правил. Этот набор правил определяет очередность файлов. Этот элемент также поддерживает счетчик, который отображает положение в медиа-файле 9. Этот счетчик в сочетании проверкой доступности файла для проигрывания и логикой правил очередности делает SEP способным интеллектуально программировать последовательность декодированных файлов, обеспечивая эффективный блок упорядочивания.

Как следует из фиг.2 и 3, при исполнении SEP 8 ядро сначала загружает информацию об очередности медиа-файла(ов) 9, который должен быть загружен, а затем начинает загружать первый кодированный файл или файл другого типа 10. Затем ядро инициализирует все необходимые медиа-системы, такие как звуковые карты или системы воспроизведения видео, и создает и/или запускает проигрыватель и цепочки загрузчика исполняемых таким образом, что оба элемента работают одновременно.

Информация об очередности синхронизована с помощью тактов (временных отрезков фиксированной длительности), каждый из которых допускает выполнение некоторого набора событий в течение такта. Такты происходят через регулярные интервалы, определенные в информации об очередности. Одним из событий в описываемой последовательности может быть изменение длительности такта в определенной точке. Это позволяет комбинировать в одной и той же композиции медиа-файлы, различающиеся по длине и ритму. События - это действия, которые могут быть выполнены проигрывателем. Некоторые из возможных действий перечислены ниже:

- начать или остановить проигрывание медиа-файла;
- изменить контент медиа-файла (напр.: размывание изображения, использование эффекта реверберации для звука);
- установить свойства медиа-файла (напр.: установить уровень тумана в 3D world [трехмерный мир виртуальной реальности])
- остановить воспроизведение;
- изменить следующий такт, который должен быть проигран (напр.: переход, повторение фрагмента);
- реагировать на ввод извне проигрывателя, например, ввод информации от пользователя или от параллельно работающей программы, или от периферийного устройства, присоединенного к компьютеру, может вызвать выполнение проигрывателем одного или более действий (событий). Ввод данных не обязательно должен происходить в той же точке, в которой событие обрабатывается проигрывателем, а введенные данные могут быть получены ранее и сохранены до тех пор, пока они не потребуются;
- синхронное управление в дополнение к событиям, которые реагируют на внешний ввод, сам проигрыватель может прямо и непосредственно отвечать на определенные команды. Такие команды могут включать способность приостанавливать проигрывание

(делать паузу) или блокировать и вновь разблокировать определенные виды событий;

- синхронизировать звуковые файлы и файлы изображения;
- синхронизировать звуковые файлы и страницы HTML;
- синхронизировать планируемую по времени доставку любого отрывка контента для Web-узла Internet.

После того, как кодированный файл или файл другого типа 10 будет загружен цепочкой загрузки, проигрыватель декодирует кодированный файл 10 в соответствующий декодированный или пригодный для проигрывания файл 19. При первой же возможности загрузчик начинает загружать следующий кодированный файл 10 в последовательности.

Тем временем проигрыватель загружает первый декодированный файл 19 [1.1 p], который требуется для начала последовательности. Проигрыватель находится в цикле, получая инструкции из информации об очередности, загруженной ядром. Если проигрыватель способен исполнить события, содержащиеся в каждом такте, в соответствии с информацией об очередности, то он их исполняет, просматривая одновременно следующий такт, чтобы убедиться в том, что события, содержащиеся в следующем такте, могут быть выполнены. Если события следующего такта не могут быть выполнены вследствие того, что следующий кодированный файл или другой тип файла 10 не был полностью загружен или декодирован, проигрыватель устанавливает последнюю точку цикла в качестве следующего такта, который должен быть выполнен, или другую точку, определенную инструкцией об очередности исполнения. После окончания одного такта проигрыватель ожидает загрузки следующего такта и повторяет этот цикл до тех пор, пока все имеющиеся такты в информации об очередности не будут исполнены над декодированными файлами 19.

В результате настоящего изобретения многие недостатки, ранее присущие передаче медиа-файлов по сетям связи, сведены к минимуму, включая проблемы передачи и воспроизведения файлов всех типов в заранее определенном порядке по сетям связи, пропускная способность которых нестабильна. В частности, некоторые из преимуществ заключаются в следующем:

а. Пользователю не требуется устанавливать специальное программное обеспечение для декодирования.

б. В отличие от потоковой передачи в реальном времени для передачи медиа-файлов и компонентов композиций, таких как музыкальные отрывки, можно использовать любую доступную полосу пропускания.

с. Пользователю не требуется ожидать предварительной загрузки всего медиа-файла или контента, такого как музыкальный отрывок или законченное речевое сообщение, прежде чем начнется проигрывание.

д. Воспроизводится подлинный предварительный записанный звук, а не звуки, генерируемые компьютером.

е. Приемник может использовать стандартное широко доступное оборудование.

ф. Любая комбинация файлов может быть представлена в виде последовательности и, таким образом, порядок загрузки файлов может быть спланирован заранее, чтобы оптимизировать ресурсы полосы пропускания.

g. Мультимедийный контент и другие сообщения могут быть построены "на лету" во время посещения с тем, чтобы избавиться от необходимости отправки полных или почти полных композиций и сделать возможным размещение более крупных композиций.

h. Факторы времени можно ввести и разместить в сообщении, таком как Web-узел, для того чтобы контент мог быть вызван в точке, обозначенной разработчиком в независимой временной шкале.

i. Композиции могут быть размещены на существующем Web-узле таким образом, чтобы связать их с конкретными страницами.

ПРИМЕРЫ

Загрузка и проигрывание сегментов данных отслеживается SEP 8 с использованием набора упорядочивающих инструкций так, что определенные резервные события инициируются в соответствии с пропускной способностью сети и тем, как это влияет на воспроизведение упомянутой последовательности. Эти функции специально сконструированы и выполняются в соответствии с инструкциями разработчика об очередности и предназначены для обеспечения следующих возможностей.

Фиг. 4 схематически показывает повторное проигрывание отдельного файла или совокупности файлов однократно или как фразу в том случае, когда следующий требуемый сегмент файла не готов для проигрывания. Применительно к Internet, примером этого может быть случай, в котором сегментный файл "а" должен быть проигран за 8 временных отрезков. Однако после 4 отрезков второй файл "b" должен проигрываться одновременно с файлом "а". В этом примере можно предположить, что файл "b" не загрузился полностью или не поступил в то место, в котором он доступен для использования. В этом случае резервное событие состоит в том, чтобы файл "а" был проигран дополнительное число "х" раз до тех пор, пока не прибьет файл "b", где "х" - это один или несколько проходов выбранного разработчиком отрезка. Как только поступает файл "b", последовательность может продолжиться.

Фиг. 5 показывает повторное проигрывание отдельного файла или совокупности файлов однократно или как фразу в том случае, когда следующий требуемый сегментный файл не готов для проигрывания. Применительно к Internet, примером этого может быть случай, в котором сегментный файл "с" должен быть проигран однократно. Однако, после того как он закончил проигрываться, второй файл "d" должен быть проигран также однократно, в определенной точке через некоторое время после файла "с" и в момент времени, заданный разработчиком. Эти два файла могут быть отдельными речевыми предложениями и могут быть неподходящими для повторения. В этом случае резервное событие состоит в том, что как только проигрывание файла "с" закончено, последовательность должна проиграть беззвучный отрезок или некоторое число

беззвучных отрезков между с & d, до тех пор, пока файл "d" не поступит и сможет быть проигран.

Фиг. 6 показывает повторное проигрывание отдельного файла или совокупности файлов однократно или как фразу в том случае, когда следующий требуемый сегмент файл не готов для повторного проигрывания, но где некоторые из файлов, включенных в повторяемый отрывок, не пригодны для повторения. Применительно к Internet, примером этого может быть случай, в котором сегментный файл "e" должен проигрываться в течение 8 временных отрезков. Однако после 4 отрезков второй файл "f" должен проигрываться вместе с файлом "e". В этом примере можно предположить, что файл "f" такого типа, который не подходит для повторного проигрывания, такого как речевое предложение, повторение которого будет выглядеть странно. Третий файл "g" должен присоединиться к файлу "e" в некоторый момент времени после проигрывания файла "f" отдельно или вместе с файлом "e". В этом примере можно предположить, что файл "g" не загрузился полностью или не поступил в нужное место, откуда он может быть использован. В этом случае резервное событие состоит в том, что файл "e" проигрывается дополнительное число "x" раз до тех пор, пока не поступит файл "g", где "x" - число, выбранное разработчиком, и где файл "f" отмечен в композиции как файл такого типа, который проигрывается только один раз и не повторяется, если некоторая часть последовательности, в которой он находится, используется повторно по какой-либо причине. Как только поступит файл "g", последовательность может продолжиться.

Фиг. 7 показывает повторное проигрывание отдельного файла или совокупности файлов однократно или как фразу в том случае, когда следующий требуемый сегмент файл не готов для повторного проигрывания, в котором резервное событие запускается в некоторый момент времени до времени проигрывания в композиции файла, который пока еще загружается. Применительно к Internet, примером этого может быть случай, в котором сегментный файл "i" должен проигрываться совместно с файлом "h". В этом примере можно предположить, что файл "i" не загрузился полностью или не поступил в нужное место, откуда он может быть использован. В этом случае резервное событие состоит в том, что по инструкции файл "h" должен быть проигран дополнительное число "x" раз с помощью специального события, помещенного внутри композиции разработчиком в некоторой точке, где повторное проигрывание файла "h" могло бы начаться в некоторое время, которое кажется более подходящим, чем после таких отрезков, например, которые были бы более привлекательны в музыкальном отношении. Как только поступит файл "i", последовательность может продолжиться.

Фиг. 8 показывает повторное проигрывание отдельного файла или совокупности файлов однократно или как фразу в том случае, когда следующий требуемый сегмент файл не готов для

повторного проигрывания, в котором резервное событие запускается в некоторый момент времени до времени проигрывания в композиции файла, который пока еще загружается. Применительно к Internet, примером этого может быть случай, в котором сегментный файл "j" is должен проигрываться в течение одного отрезка. Однако еще через 2 отрезка второй файл "k" также должен проигрываться. В этом примере можно предположить, что файл "k" не загрузился полностью или не поступил в нужное место, откуда он может быть использован. В этом случае резервное событие состоит в том, что по инструкции файл "j" не должен проигрываться до тех пор, пока файл "k" тоже не загрузится, что обеспечивается специальным событием, размещенным внутри композиции разработчиком. Это событие должно предварительно выбрать файл "k" или до начала последовательности, или же, если эта ситуация возникла позднее в данной композиции, продолжить последовательность таким образом, чтобы обеспечить между этими двумя файлами интервал времени, который установлен разработчиком и управляется тактами синхронизации последовательности. Как только поступит файл "k", последовательность может продолжиться путем проигрывания файлов "j" и "k".

Фиг. 9 показывает проигрывание и повторное проигрывание отдельного файла или совокупности файлов, как показано в вышеприведенных примерах, но где композиция приостанавливается или заканчивается на заранее определенное время в точке, выбранной разработчиком, а затем продолжается либо путем повторного использования уже загруженных файлов или проигрывания новых файлов, или некоторой комбинации этих двух способов. Применительно к Internet, примером этого может быть случай, в котором часть композиции или последовательности проиграна до точки, с которой разработчиком предусмотрена пауза в течение некоторого времени. Здесь время регулируется путем вставки соответствующего числа беззвучных отрезков известной длительности или путем вставки в композицию специально подобранного отрезка известной длительности. Следующий пакет сегментированных данных приходит до времени, установленного разработчиком для проигрывания, с тем, чтобы оптимизировать исполнение композиции. Это событие позволяет разработчику ввести понятие времени, например страница Internet аудиофайл вслух объявляет, что некоторое событие произойдет через некоторый промежуток времени.

Синхронизация данных разного типа, таких как файлы изображений, документы HTML, анимированные изображения, другие исполняемые файлы (другие апплеты), инструкции для других устройств, потоки данных и аудиофайлы, где любое из вышеупомянутых резервных событий может произойти так, чтобы сделать возможной доставку композиции/последовательности по такой сети как Internet, где характеристики пропускной способности /полосы пропускания сети подвержены колебаниям. Применительно к рекламе в Internet,

примером синхронизации файлов разных типов может быть затягивание аудиопоследовательности в HTML Web-страницы. Упомянутая Web-страница располагается на узле с тем, чтобы она сохранялась в течение всего визита на узел. Внутри последовательности содержатся совокупность файлов изображений, а также аудиофайлы, которые создают фоновый звук. Эти изображения представляют собой файлы изображений, которые позже должны быть использованы на следующей странице HTML. Эта вторая страница HTML также помещается внутри данной последовательности и запуск этой страницы управляется планировщиком так, что он не происходит до тех пор, пока не будут полностью загружены все файлы изображений и любые другие, необходимые для нее файлы. В нужный момент времени, когда все необходимые файлы уже загружены, планировщик переходит к той точке, где он запускает вторую страницу HTML и направляет эту страницу в предварительно выбранную область экрана. Результатом этого является автоматическая синхронизация доставки страницы HTML с проигрыванием и показом содержащихся в ней аудиофайлов и изображений. Так как этот способ управляется исходной последовательностью, то может произойти переключение обратно к предыдущей странице или произведена синхронизация третьей страницы тем же способом. Целью здесь является использование этого механизма для создания нового метода рекламы в Internet, который не только является аудио/визуальным, но может возобновляться и снова приостанавливаться подобно рекламной паузе в телевидении.

Инструкция очередности может также содержать инструкции, приводящие к повторам переменной длительности или повторам фиксированной длительности и переходам, при которых данные, проигрываемые в повторяемом фрагменте, уже загружены или представляют собой комбинацию уже загруженных и новых данных. Повторы переменной длительности могут быть типа DS, Fine, Da Capo, Coda, и т.д., которые встречаются в музыкальных сочинениях. Другим примером повторов переменной длительности может быть использование этого механизма для доставки анимации, в которой фоновое изображение при первом проигрывании отличается от того, которое используется в повторяющемся отрывке, а изображения первого плана одинаковы и используются повторно.

Временная характеристика последовательности может также варьировать и изменяться в разных местах композиции. Целью этого является регулировать скорость, с которой проигрываются файлы, дать возможность использовать файлы разных музыкальных стилей внутри одной композиции, создать точки проигрывания на пути проигрывания файла путем деления времени последовательности на выбранное число, дать возможность повторного проигрывания контента в заданной точке будущего сообщения и тому подобное.

Хотя настоящее изобретение описано выше на предпочтительном варианте его осуществления, специалистам в данной

области техники очевидны любые его изменения и модификации, которые могут быть внесены в пределах существа настоящего изобретения. Все такие изменения и модификации следует рассматривать как включенные в объем данного изобретения.

Формула изобретения:

1. Способ передачи медиа-файла по сети связи и воспроизведения или проигрывания упомянутого файла, включающий: (a) разделение упомянутого медиа-файла на последовательность сегментированных файлов, (b) поддержание упомянутых сегментированных файлов в средствах компьютера провайдера и (c) передача, в определенной последовательности, упомянутых сегментированных файлов к средствам принимающего компьютера, в котором после того как каждый сегмент был получен упомянутым принимающим компьютером, проигрывание или воспроизведение сегмента начинается до или во время загрузки следующего сегмента последовательности, и, кроме того, в котором последующие полученные сегменты проигрываются в соответствии с информацией об очередности, в котором информация об очередности определяет, проигрывается ли какой-либо сегмент, какой именно и каким образом, в том случае, когда следующий сегмент последовательности не доступен для проигрывания или воспроизведения.

2. Способ по п. 1, отличающийся тем, что стадия (b), кроме того, включает поддержание загружаемой пользователем программы в средствах компьютера провайдера и стадия (c), кроме того, включает передачу этой программы к средствам принимающего компьютера.

3. Способ по п. 1, отличающийся тем, что данный сегмент кодируется и декодируется загружаемой пользователем программой в средствах принимающего компьютера до проигрывания или воспроизведения.

4. Способ по п. 1, отличающийся тем, что данная программа является Java-апплетом или ActiveX файлом.

5. Способ по п. 1, отличающийся тем, что проигрывание медиа-файла может начинаться до завершения процесса получения следующего сегмента последовательности.

6. Способ по п. 1, отличающийся тем, что информация об очередности параметризована и передается в текстовом документе или в файле другого подобного типа.

7. Способ по п. 6, отличающийся тем, что информация об очередности встроена в загружаемую пользователем программу.

8. Способ по п. 1, отличающийся тем, что медиа-файл представляет собой любую совокупность данных, таких, как аудиофайл, видеофайл, файл изображения, и файл анимационного изображения, HTML файл, VRNML/3D World файл, текстовый файл, Flash фильм, другие используемые файлы, инструкции для других устройств, потоки данных или фильтр, который модифицирует другие медиа.

9. Система принимающего компьютера, включающая средства для воспроизведения или проигрывания медиа-файла, переданного

по сети связи от компьютерных средств провайдера, располагающих средствами для разделения и поддержания медиа-файла в виде последовательности сегментов и переноса сегментов в специфической последовательности по сети связи к средствам принимающего компьютера, в которых, после того как каждый из упомянутых сегментов был получен упомянутым принимающим компьютером, он

будет декодирован и проигрывание или воспроизведение упомянутых сегментов может начаться до или во время загрузки следующего сегмента последовательности, таким образом, что декодированные файлы могут быть воспроизведены или проиграны способом идентичным тому, каким была построена композиция до того, как произошло любое резервное событие в воспроизведении.

5

10

15

20

25

30

35

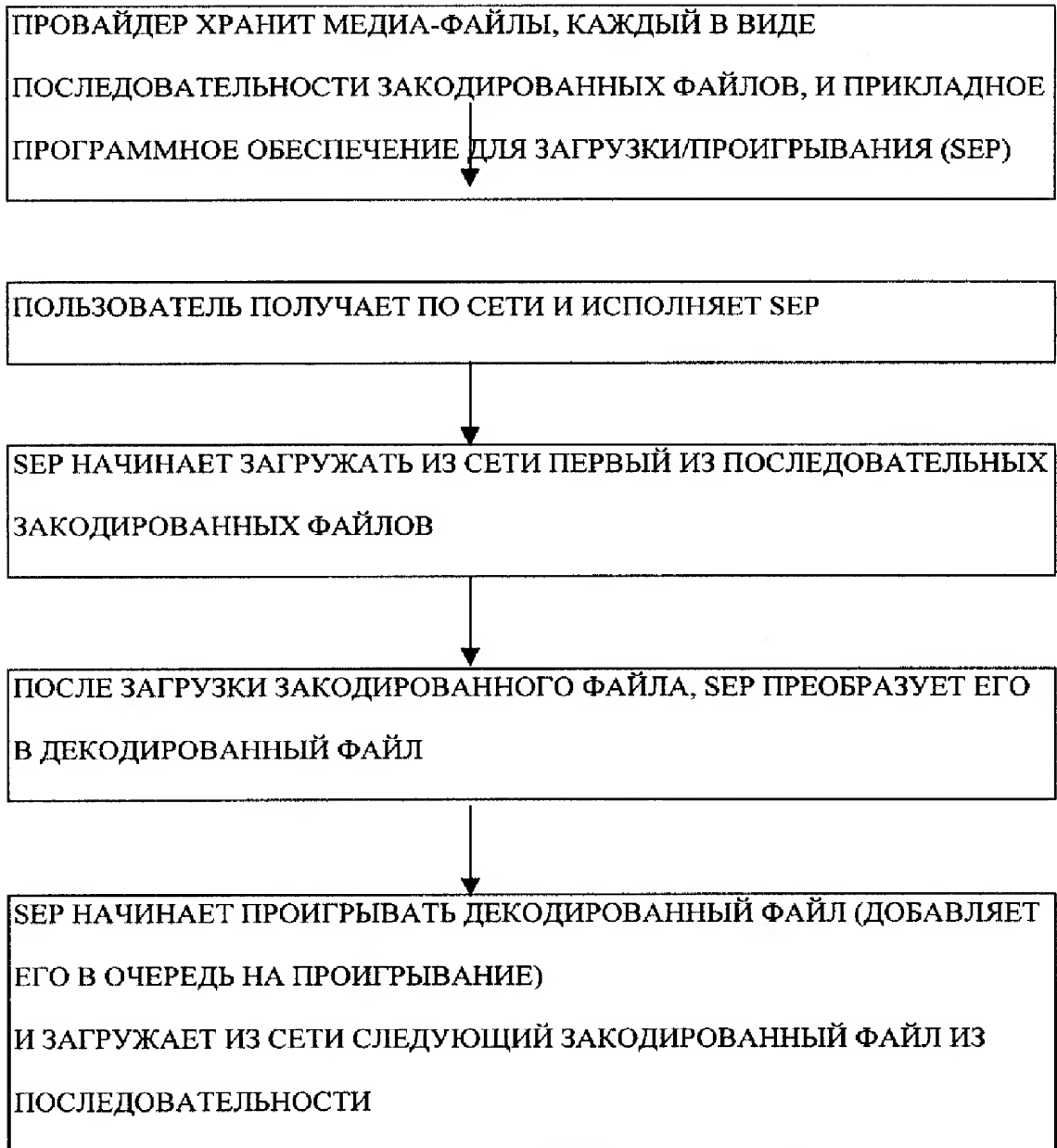
40

45

50

55

60



Фиг. 2

Псевдокод ЯДРА

```
{ Init}
    Загрузить информацию о последовательности, включая список медиа-файлов
    Начать загрузку медиа-файлов, требующихся для начала последовательности
    Ждать пока не загрузятся все вышеупомянутые файлы
    Инициализировать все требуемые медиа системы (напр.: воспроизведение звука или видео)
    Создать и начать ПРОЦЕСС ПРОИГРЫВАТЕЛЯ (смотря сразу ниже)
    Начать ПРОЦЕСС ЗАГРУЗЧИКА (смотри ниже)
{ Конец Init}
```

Псевдокод для ПРОЦЕССА ПРОИГРЫВАТЕЛЯ

```
{ Основной код}
    FOR EACH такт информации о последовательности DO
        FOR EACH событие в такте DO
            исполнить событие (это может включать проигрывание звукового файла,
            остановку воспроизведения, изменение такта, который должен
            исполняться
            следующим, и т.д.)
        конец FOR EACH

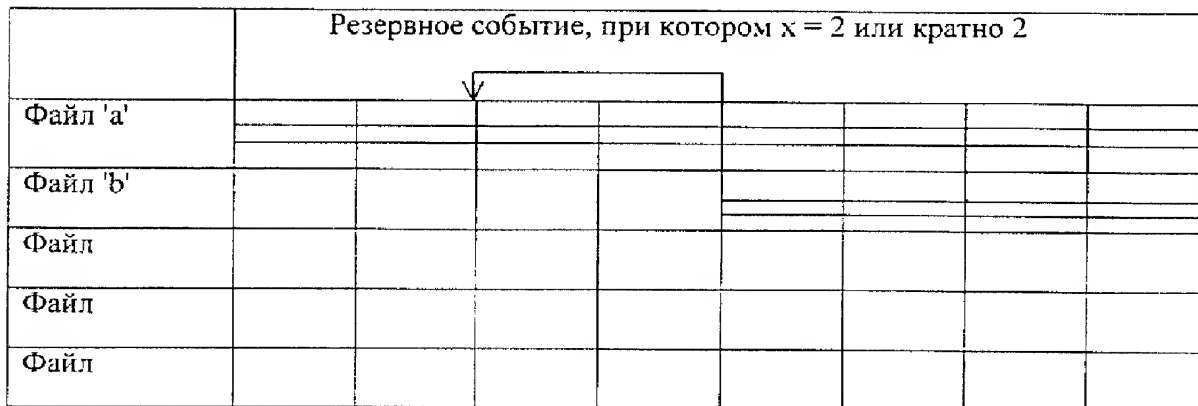
        анализ следующего такта, убедиться что все события данного такта
        готовы для исполнения

        если это не так
            установить последний встреченный такт в качестве следующего такта для
            исполнения
            возврат к началу цикла
        ожидать пока не настанет время начать проигрывание следующего такта
        последовательности
    конец FOR EACH
{ Конец основного код}
```

Псевдокод для ПРОЦЕССА ЗАГРУЗЧИКА

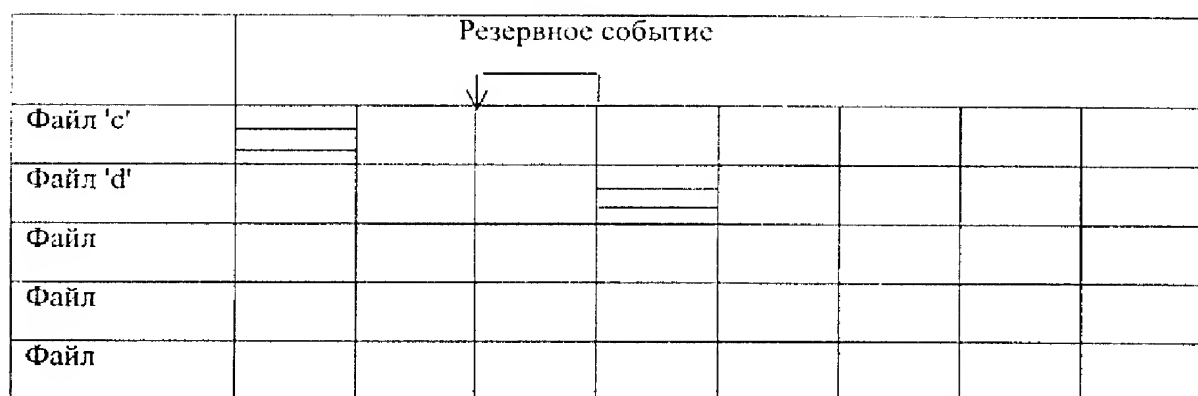
```
{ Основной код}
    FOR EACH медиа-файл DO
        Если загрузка файла не началась
            начать файл загрузку файла
        если загрузка файла не закончилась
            ждать пока не закончится загрузка файла
        конец FOR EACH
{ Конец основного кода}
```

Фиг. 3



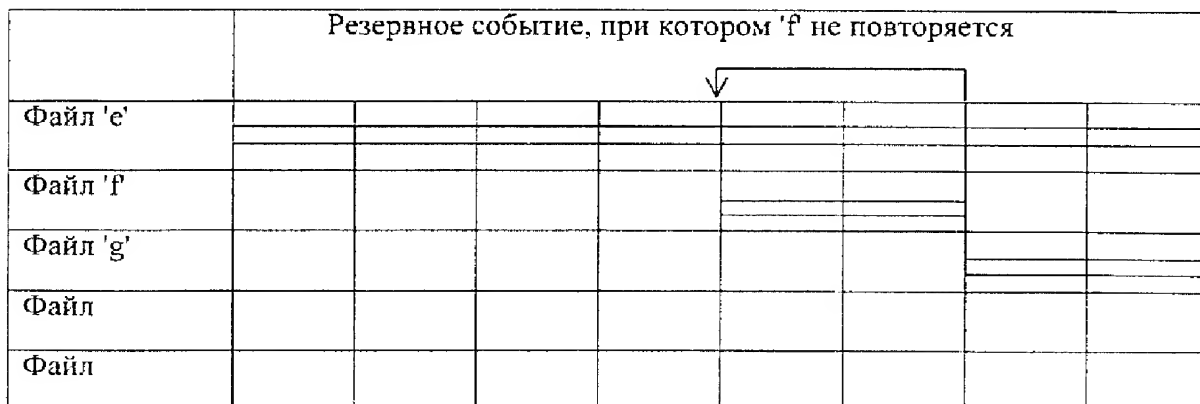
----- Отрезки времени -----

Фиг. 4



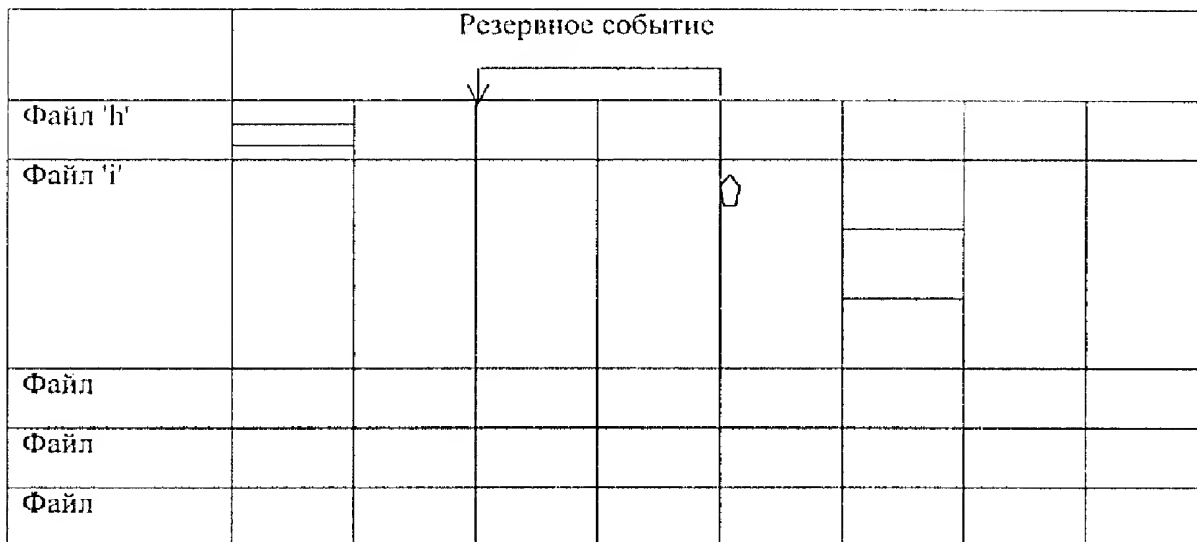
----- Отрезки времени -----

Фиг. 5



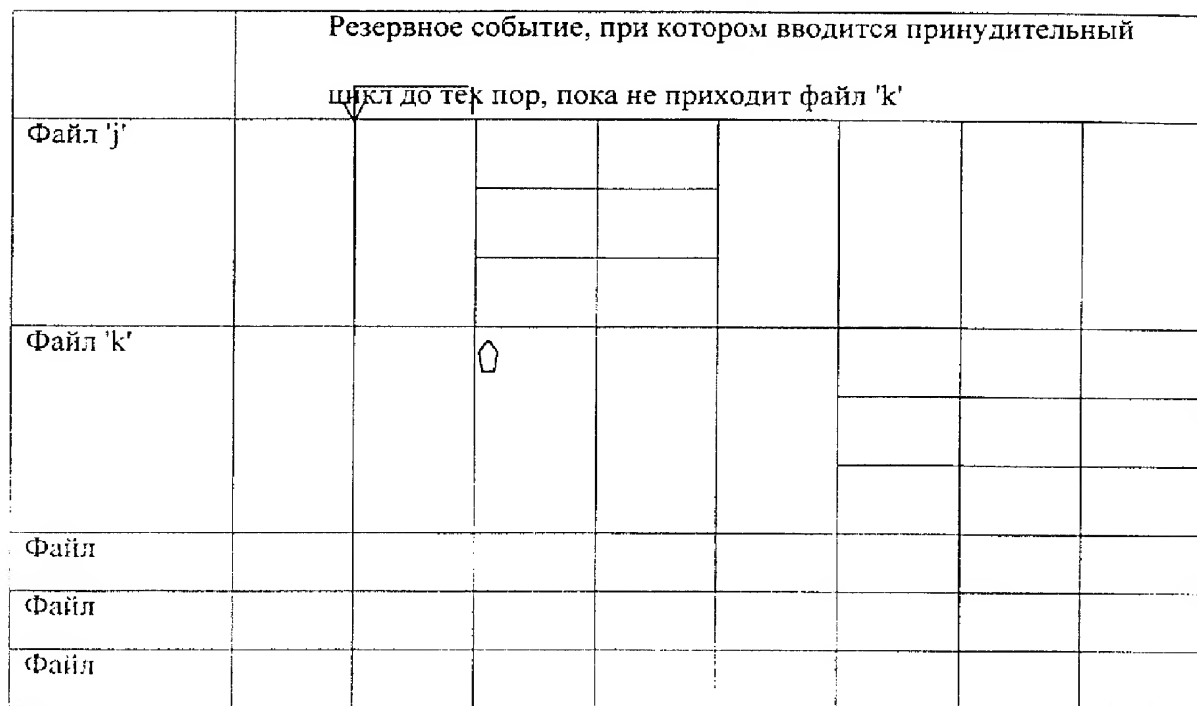
----- Отрезки времени -----

Фиг. 6



----- Отрезки времени -----

Фиг. 7



----- Отрезки времени -----

Фиг. 8

	Резервное событие, при котором 'ТС' указывает изменение скорости исполнения последовательности							
Файл 'l'								
Файл 'm'								
Файл 'n'								
Файл 'o'								
Файл								

----- Отрезки времени -----

Фиг. 9

RU 2189119 C2

RU 2189119 C2